

Notice of Allowability

Application No.

09/184,738

Examiner

Todd Ingberg

Applicant(s)

MORRIS ET AL.

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 11/28/2005.
2. ☒ The allowed claim(s) is/are 1,4-11,16,20-22,25-28,31-38,43,47-56,58-75.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☐ All b) ☐ Some* c) ☐ None of the:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☒ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☒ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☒ hereto or 2) ☐ to Paper No./Mail Date _____.
 - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date 1/9/2005.
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____

Todd Ingberg
Primary Examiner
Art Unit: 2193

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR

1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Laurence Weinberger on January 9, 2005.

The application has been amended as follows:

Marked Copy

1. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

[a.] **a)** means for dynamically wrapping objects written to a known specification, with additional properties beyond those properties internal to the objects; and

[b.] **b)** means for utilizing the additional **and/or internal** properties to modify the execution of the objects.

4. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

a) an interpreting run time program;

b) scripts that contain property settings for the objects; and

c) means for utilizing objects by specifying property values according to the script

8. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

a) an interpreting run time program that has no logical or arithmetic operators;

b) scripts that contain property settings for the objects; and

e) means for utilizing objects by specifying property values according to the script.

Art Unit: 2193

16. (currently amended) The system of claim 1, 4, 5, 6, 7, 8, 9, 10, or 11 further comprising a means for adding additional programming constructs or sub-programs by employing objects that perform the function of programming constructs **or sub-programs** [utilizing objects] wherein expansion of program capabilities is achieved.

20. (currently amended) A computer implemented development and run time system **employing a central processing unit with access to memory and storage** employing one or more objects written to a known specification which utilizes one or more scripts in which both the

objects and the scripts can be maintained separately, utilizing a set of core functionalities comprising:

[a.] **a**) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and

b) means for instantiating objects;

c) means for integrating objects;

d) means for sequencing objects; and

e) means for providing communication among objects wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

21. (currently amended) A computer implemented run time system **employing a central processing unit with access to memory and storage** employing a run time program, one or more objects written to a known specification utilizing a set of core functionalities which interprets one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and

b) means for instantiating objects;

c) means for integrating objects;

d) means for sequencing objects;

e) means for providing communication among objects

wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

22. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** for employing one or more objects written to a known specification, having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising:

a) means for setting the values of properties and connecting events;

b) means for recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and

c) means for traversing the history one change at a time wherein the property values and event connections may be edited from any point in the history.

Art Unit: 2193

25. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** [which] that interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects written to a known specification, which can be maintained separately, provides and manages data flow among objects, and manages the execution of objects comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and
- b) means for utilizing the additional **and/or internal** properties to manage the execution of the objects and the additional **and/or internal** events to link and sequence the objects wherein the run time execution of the objects is determined by property values and events.

26. (currently amended) A computer implemented system **employing a central process unit with access to memory and storage** employing two or more objects written to a known specification which implements parallel processing comprising:

- a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events internal to the objects;
- b) means for utilizing the additional **and/or internal** properties to manage the execution of the objects and the additional **and/or internal** events to link and sequence the objects; and
- c) means for specifying the temporal relationship among objects by placing the objects on one or more time lines wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

27. (currently amended) An object oriented programming computer implemented system **employing a central processing unit with access to memory and storage** in which the function of programming constructs is achieved by utilizing objects written to a known specification comprising:

- a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;
- b) means for utilizing the additional **and/or internal** properties and events to link and sequence the objects; and
- c) means for specifying a list of property values and event settings wherein the execution of the objects is determined by the list of property values and event settings.

28. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising **the steps of:**

- a) dynamically wrapping objects written to a known specification with additional properties beyond those properties internal to the object; and
- b) utilizing the additional **and/or internal** properties to modify the execution of the objects.

31. (currently amended) A computer implemented software method for interpreting

application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising **the steps of:**

- a) utilizing an interpreting run time program;
- b) **utilizing** scripts that contain property settings for the objects; and
- c) utilizing objects by specifying property values according to the script.

35. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising **the steps of:**

- a) utilizing an interpreting run time program that has no logical or arithmetic operators;
- b) **utilizing** scripts that contain property settings for the objects; and
- c) utilizing objects by specifying property values according to the script.

48. (currently amended) A computer implemented run time software method employing one or more objects written to a known specification for executing an application utilizing a set of core functionalities which interprets one or more scripts, in which [both] **the run time program, the objects and the script** can be maintained separately, comprising the steps of

- a) instantiating objects;
- b) integrating objects;
- c) sequencing objects; and
- d) providing communication among objects wherein the functionalities performed by the software method during execution are determined by the objects used and the scripts.

50. (currently amended) A computer implemented run time software method employing objects which interprets a script containing property values and event settings, in which [both] **the run time program, the objects and the script** can be maintained separately, and dynamically executes the objects comprising the steps of;

- a) wrapping objects with additional properties and events beyond those properties and events internal to the objects;
- b) utilizing the additional **and/or internal** properties and events to link and sequence the objects; and
- c) reading one or more sets of property values and event settings maintained separately from the run time system and the objects wherein the execution of the objects is determined by the property values and event settings in the script.

52. (currently amended) A computer implemented software method [which] **that** interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects written to a known specification, **which can be maintained separately**, provides and manages data flow among objects, and manages the execution of objects comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and

Art Unit: 2193

b) utilizing the additional **and/or internal** properties to manage the execution of the objects and the additional **and/or internal** events to link and sequence the objects wherein the execution of the objects is determined by the property values and events.

53. (currently amended) A computer implemented software method employing two or more objects written to a known specification which implements parallel processing comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events [provide] internal to the object;
- b) utilizing the additional **and/or internal** properties **to manage the execution of the objects** and the additional **and/or internal** events to link and sequence the objects; and
- c) specifying the temporal relationship among objects by placing the objects on one or more time lines wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

54. (currently amended) A computer implemented object oriented software programming method in which the function of programming constructs is achieved by utilizing objects written to a known specification comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;
- b) utilizing the additional **and/or internal** properties and events to link and sequence the objects; and
- c) specifying a list of property values and event settings wherein the execution of the objects is determined by the list of property values and event settings.

55. (currently amended) A computer implemented software method for employing one or more standardized objects written to a known specification with properties not internal to the standardized objects comprising the steps of:

- a) using a wrapper object, dynamically wrapping [the] **one or more** standardized [object] **objects** with additional properties beyond those properties internal to the one or more standardized [object] **objects**; and
- b) utilizing the additional **and/or internal** properties to control the standardized [object] **objects**.

56. (currently amended) A computer implemented software method for employing one or more standardized objects written to a known specification with events not internal to the standardized objects comprising the steps of

- a) using a wrapper object, dynamically wrapping [the] **one or more** standardized [object] **objects** with additional events beyond those events internal to the **one or more** standardized objects; and
- b) utilizing the additional **and/or internal** events to control the **one or more** standardized [object] **objects**.

Art Unit: 2193

58. (currently amended) A computer implemented system **employing a central processing unit with access to memory and storage** for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

[a.] **a** means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and

[b.] **b** means for utilizing the additional **and/or internal** events to link and sequence the objects,

60. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts in which the run time program, the objects, and the scripts can be maintained separately, comprising **the steps of:**

a) using a wrapping object, dynamically wrapping the object written to a known specification with additional events beyond those events internal to the object; and

b) utilizing the additional **and/or internal** events to link and sequence the object.

61. (currently amended) A computer implemented run time program method to employ a one or more objects written to a known specification with properties, not native to the objects, specified by one or more scripts comprising the steps of:

a) using an object [internal] **available** to the run time program, dynamically wrapping the object written to a known specification with additional properties; and

b) utilizing the additional **and/or internal** properties to modify the execution of the object written to a known specification.

65. (currently amended) A computer implemented run time program method to employ one or more objects written to a known specification with events, not native to the objects, specified by one or more scripts comprising the steps of:

[b.] **a** using an object [internal] **available** to the run time program, dynamically wrapping the object written to a known specification with additional events; and

b) utilizing the additional **and/or internal** events to link and sequence the object written to a known specification.

69. (currently amended) A computer implemented method for interpreting one or more application scripts employing a runtime program comprising the steps of:

a) loading one or more application scripts that can be maintained separately from the runtime program;

b) according to information stored in the application script, loading one or more objects that can be maintained separately from the runtime program and the scripts;

c) using an object [internal] **available** to the run time program, dynamically wrapping one or more objects written to a known specification with additional properties beyond those properties internal to the object; and

d) utilizing the additional **and/or internal** properties [added to objects] to modify the execution of the objects based on the values found in the script.

Clean Copy

1. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping objects written to a known specification, with additional properties beyond those properties internal to the objects; and
- b) means for utilizing the additional and/or internal properties to modify the execution of the objects.

4. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

- a) an interpreting run time program;
- b) scripts that contain property settings for the objects; and
- c) means for utilizing objects by specifying property values according to the script

8. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

- a) an interpreting run time program that has no logical or arithmetic operators;
- b) scripts that contain property settings for the objects; and
- e) means for utilizing objects by specifying property values according to the script.

16. (currently amended) The system of claim 1, 4, 5, 6, 7, 8, 9, 10, or 11 further comprising a means for adding additional programming constructs or sub-programs by employing objects that perform the function of programming constructs or sub-programs wherein expansion of program capabilities is achieved.

20. (currently amended) A computer implemented development and run time system employing a central processing unit with access to memory and storage employing one or more objects written to a known specification which utilizes one or more scripts in which both the objects and the scripts can be maintained separately, utilizing a set of core functionalities comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and
- b) means for instantiating objects;
- c) means for integrating objects;
- d) means for sequencing objects; and

Art Unit: 2193

c) means for providing communication among objects wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

21. (currently amended) A computer implemented run time system employing a central processing unit with access to memory and storage employing a run time program, one or more objects written to a known specification utilizing a set of core functionalities which interprets one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and
- b) means for instantiating objects;
- c) means for integrating objects;
- d) means for sequencing objects;
- e) means for providing communication among objects wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

22. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for employing one or more objects written to a known specification, having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising:

- a) means for setting the values of properties and connecting events;
- b) means for recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and
- c) means for traversing the history one change at a time wherein the property values and event connections may be edited from any point in the history.

25. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage that interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects written to a known specification, which can be maintained separately, provides and manages data flow among objects, and manages the execution of objects comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and
- b) means for utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects wherein the run time execution of the objects is determined by property values and events.

26. (currently amended) A computer implemented system employing a central process unit with access to memory and storage employing two or more objects written to a known specification which implements parallel processing comprising:

- a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events internal to the objects;

Art Unit: 2193

b) means for utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects; and

c) means for specifying the temporal relationship among objects by placing the objects on one or more time lines wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

27. (currently amended) An object oriented programming computer implemented system employing a central processing unit with access to memory and storage in which the function of programming constructs is achieved by utilizing objects written to a known specification comprising:

a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;

b) means for utilizing the additional and/or internal properties and events to link and sequence the objects; and

c) means for specifying a list of property values and event settings wherein the execution of the objects is determined by the list of property values and event settings.

28. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising the steps of:

a) dynamically wrapping objects written to a known specification with additional properties beyond those properties internal to the object; and

b) utilizing the additional and/or internal properties to modify the execution of the objects.

31. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run tune program, the objects and the scripts can be maintained separately, comprising the steps of:

a) utilizing an interpreting run time program;

b) utilizing scripts that contain property settings for the objects: and

c) utilizing objects by specifying property values according to the script.

35. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising the steps of:

a) utilizing an interpreting run tune program that has no logical or arithmetic operators;

b) utilizing scripts that contain property settings for the objects; and

c) utilizing objects by specifying property values according to the script.

48. (currently amended) A computer implemented run time software method employing one or more objects written to a known specification for executing an application utilizing a set of core

Art Unit: 2193

functionalities which interprets one or more scripts, in which the run time program, the objects and the script can be maintained separately, comprising the steps of

- a) instantiating objects;
- b) integrating objects;
- c) sequencing objects; and
- d) providing communication among objects wherein the functionalities performed by the software method during execution are determined by the objects used and the scripts.

50. (currently amended) A computer implemented run time software method employing objects which interprets a script containing property values and event settings, in which the run time program, the objects and the script can be maintained separately, and dynamically executes the objects comprising the steps of;

- a) wrapping objects with additional properties and events beyond those properties and events internal to the objects;
- b) utilizing the additional and/or internal properties and events to link and sequence the objects; and
- c) reading one or more sets of property values and event settings maintained separately from the run time system and the objects wherein the execution of the objects is determined by the property values and event settings in the script.

52. (currently amended) A computer implemented software method that interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects written to a known specification, which can be maintained separately, provides and manages data flow among objects, and manages the execution of objects comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and
- b) utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects wherein the execution of the objects is determined by the property values and events.

53. (currently amended) A computer implemented software method employing two or more objects written to a known specification which implements parallel processing comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the object;
- b) utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects; and
- c) specifying the temporal relationship among objects by placing the objects on one or more time lines wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

Art Unit: 2193

54. (currently amended) A computer implemented object oriented software programming method in which the function of programming constructs is achieved by utilizing objects written to a known specification comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;
 - b) utilizing the additional and/or internal properties and events to link and sequence the objects; and
 - c) specifying a list of property values and event settings
- wherein the execution of the objects is determined by the list of property values and event settings.

55. (currently amended) A computer implemented software method for employing one or more standardized objects written to a known specification with properties not internal to the standardized objects comprising the steps of:

- a) using a wrapper object, dynamically wrapping [the] one or more standardized objects with additional properties beyond those properties internal to the one or more standardized objects; and
- b) utilizing the additional and/or internal properties to control the standardized objects.

56. (currently amended) A computer implemented software method for employing one or more standardized objects written to a known specification with events not internal to the standardized objects comprising the steps of

- a) using a wrapper object, dynamically wrapping one or more standardized objects with additional events beyond those events internal to the one or more standardized objects; and
- b) utilizing the additional and/or internal events to control the one or more standardized objects.

58. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and
- b) means for utilizing the additional and/or internal events to link and sequence the objects,

60. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification, and one or more scripts in which the run time program, the objects, and the scripts can be maintained separately, comprising the steps of:

- a) using a wrapping object, dynamically wrapping the object written to a known specification with additional events beyond those events internal to the object; and
- b) utilizing the additional and/or internal events to link and sequence the object.

Art Unit: 2193

61. (currently amended) A computer implemented run time program method to employ a one or more objects written to a known specification with properties, not native to the objects, specified by one or more scripts comprising the steps of:

a) using an object available to the run time program, dynamically wrapping the object written to a known specification with additional properties; and

b) utilizing the additional and/or internal properties to modify the execution of the object written to a known specification.

65. (currently amended) A computer implemented run time program method to employ one or more objects written to a known specification with events, not native to the objects, specified by one or more scripts comprising the steps of:

a) using an object available to the run time program, dynamically wrapping the object written to a known specification with additional events; and

b) utilizing the additional and/or internal events to link and sequence the object written to a known specification.

69. (currently amended) A computer implemented method for interpreting one or more application scripts employing a runtime program comprising the steps of:

a) loading one or more application scripts that can be maintained separately from the runtime program;

b) according to information stored in the application script, loading one or more objects that can be maintained separately from the runtime program and the scripts;

c) using an object available to the run time program, dynamically wrapping one or more objects written to a known specification with additional properties beyond those properties internal to the object; and

d) utilizing the additional and/or internal properties to modify the execution of the objects based on the values found in the script.

76. (new) A computer implemented method employing a runtime program for interpreting one or more application scripts comprising the steps of:

a) loading from a remote information source one or more application scripts that can be maintained separately from the runtime program;

b) loading from a remote information source, from locations specified by the one or more scripts, objects that can be maintained separately from the runtime program and the one or more scripts; and

c) utilizing properties values in the one or more scripts to modify the execution of the objects

REASONS FOR ALLOWANCE

2. The following is an examiner's statement of reasons for allowance:

The following outlines the reasons for allowance.

- I. Terms
- II. Closest Art of Record
- III. Amended Claims
- IV. Examiner's Comment's

I. Terms

The response to the section Interview Summary clarifies the key terms (e.g. object and wrapping, their meaning use and function and the environments etc) in the Application.

The response reads as follows:

“

REMARKS/ARGUMENTS

INTERVIEW SUMMARY

During a very productive discussion with the Examiner, several key points of the invention were discussed which have a direct bearing on the claims. Before proceeding to respond to the issues raised by the Examiner in the Office Action dated July 29, 2005, Applicants will summarize these points since they will serve as the basis for the Response.

Initially Applicants wanted to clarify the definition of "object" since the Examiner had raised the issue in the Office Action and an understanding of "object" is central to the understanding of Applicants' invention. The Examiner's attention was drawn to the definition of "object" contained in the specification. Applicants submit that within the definition provided by the specification, those skilled in the art will understand an object to be an encapsulated executable which can be instantiated and operate on supplied parameters. The specification discusses the use of objects written to some known specification and gives an example of VBX objects.

During the interview, Applicants pointed out that at the time of Applicants' invention, the prior art taught that objects could, during their construction, have written into them any number of properties and events, and it was common practice at that time for developers to use one or more class libraries to create a set of compatible objects. However, it was not appreciated how to add additional properties and events to a third party objects without rewriting the objects. The invention of the present application adds properties and events to existing third party objects without rewriting the objects. As disclosed in the application, the addition of properties and events is accomplished by wrapping the objects within instances of a wrapping object.

A second feature of applicants' invention that was discussed is the dynamic (real time) wrapping of objects with additional properties and events by both the development and the

Art Unit: 2193

runtime program as the programs are running. Initially, the objects with their built-in (native) properties and events reside in storage locations accessible to the programs of the invention. At this point, the objects have no additional properties and events added. Then, when the development and runtime programs of the invention require an object, the object is queried to determine its native properties and events, and the object is dynamically wrapped to add the additional properties and events that are used to control the object. In the exemplary program disclosed in the application, a single set of additional properties and events are added to each wrapped object. However, as discussed with the Examiner, different objects may be wrapped with different properties and events if that is required. The wrapping is a dynamic process and is accomplished on-the-fly. No permanent repository of the wrapped objects is kept.

Also discussed was an aspect of the invention that results from the objects being dynamically wrapped, namely the fact that the objects can be located anywhere as long as the location of the object is provided to the program (either development or runtime) and the object can be accessed. The script contains not only a listing of properties and events, but also contains the storage location for each object used by the script. The script is a separate file that has its own location. The script's location can be provided to the program by several methods well known in the art, for example by a command line argument, or by another script, or by dragging and dropping. Similar to objects, the script may also be located anywhere as long as the location of the script is provided to the program (either development or runtime) and the script can be accessed. Consequently, it is not necessary for the development program, the runtime program, the scripts, or the objects to be physically located near one another. It was pointed out to the Examiner that this is the basis for the "maintained separately" language relating to the objects and scripts recited in the claims.

Finally, the Applicants discussed with the Examiner the generality achieved by the construction of Applicants' invention. The invention places no limits upon the functions contained within the objects which are used and no limits on the abilities of the objects to interact with systems outside the invention's programs. For instance, as one type of example, it was pointed out that the program implements logical operators and looping constructions, variables, conditionals, and other features by employing objects which perform those functions. Similarly, objects implementing any other imaginable function can be used with the invention. Any type of program and/or program construct can be formulated with Applicants' invention. Examples of parallel processing programs, robotics, and language translation programs as being possible to implement with Applicants' invention were mentioned during the interview, but are by no means meant to limit the generality of the possible uses of Applicants' invention.

The general features of the invention discussed above are reflected in the claims. During the interview, Applicants provided the Examiner, at the Examiner's request, with a DVD version of the VHS video tape that had been in the Patent Office file of the parent application (issued as U.S. Patent No. 5,862,372) but which could not be located by the Examiner. The VHS video tape had been previously supplied for review as an aid in helping the examiner of the parent application understand the program. In addition, Applicants briefly displayed a working/running version of the program and showed the Examiner how programs were constructed with the invention.

During an earlier telephonic interview and again at the October Interview, Applicants' attorney pointed out to the Examiner that Applicants had become aware that the software code

Art Unit: 2193

appendix supplied on CD-ROM as part of the application was missing a single header file (workform.h) from the initial parent application program. (The header file is listed in the file list contained on the CD-ROM, but, for reasons unknown to the inventors, the actual file failed to copy to the CD-ROM during duplication). In the parent application, the entire code had been supplied as a several hundred page printout. During prosecution of the parent, Applicants were requested to file the code as a CD-ROM appendix and did so. A copy of that CD-ROM was filed with the present application. The header file that is missing from the CD-ROM does not affect the scope of the claims or the disclosure. This file is necessary to compile the program (since there is an internal reference to the file) but it can be easily generated by those skilled in the art from information contained within the program code without undue experimentation. Applicants submit that the code provided on CD-ROM provides a sufficiently complete written description of the invention and is enabling of the claims. The Examiner indicated that he understood what had happened and that he did not believe it presented any written description or enablement problem for the application.

The paper copy printout of the entire program code is still maintained in the original application. Attached to this Response is a copy of the missing header file copied from the program code printout submitted in the parent application. It is placed in the record of this application as a convenience to anyone desiring it who does not wish to go to the effort of constructing the header file themselves.”

II. Closest Art of Record

The closest art of record was applied in the Office Action dated July 29, 2005. The Star reference was persuasively argued by Applicant in their response. The persuasive is located on pages 23 to 47 for the claims rejected in the Office action.

III. Amended Claims

The Examiner made an Examiner's amendment to clarify the tangibility of the invention and to correct unintentional and inadvertent errors in the Response to Office Action.

“

REMARKS

Applicants have amended the claims to achieve two purposes. First, all the system claims have been amended to reflect integration with computer hardware. Antecedent basis for the amendments is found in Figure 1 and the following paragraph from the specification:

Figure 1 is a block diagram showing the basic functional units of a modern digital computer with which the computer implemented applications development system of the invention works. Information may be stored on magnetic or optical media for access or be available through a connection to a remote information source. During the development of an application with the system of this invention, the user directs the incorporation of objects into one of the four views, modifies the properties associated with that object, and interconnects the objects so chosen. The user may direct the machine to accomplish these tasks using a pointer (such as a mouse) or a keyboard and visually observes his/her actions on the system's output display. Second, in those cases where there are substantially parallel system and method claims, the claims have been amended to more closely align the claims. The following is a brief summary of these changes. Claim 16 was amended to add the language "or sub-programs" found

Art Unit: 2193

in claim 43. Claim 52 was amended to add the language "which can be maintained separately" after reference to the objects as found in claim 25. Claim 53 was amended to add the language "to manage the execution of the objects" as found in claim 26.

Further amendments were made to claims which recited that the "additional properties" or the "additional events" were utilized. Once Applicants' method has wrapped objects with additional properties and/or events, the method can use both the native internal properties and events as well as the additional properties and events to manage, sequence, and Link the objects. Accordingly, the reference to the "additional properties" or the "additional events" was amended to "additional and/or internal" to clearly recite that both the additional and the native internal properties and events may be used.. Claims 1, 27, 25, 26, 27, 28, 50, 52, 53, 54, 55, 56, 58, 60, 61, and 65 were so amended. Claim 69 was amended for a similar reason.

Claims 61, 65, and 69 were amended to recite "available" in place of "internal." The wrapping object only needs to be available for use by the run time program and need not be internal to the run time program. Applicants' program uses a wrapping object internal to the program, but once the use of a wrapping object is appreciated as has been taught by Applicants, Applicants submit that those skilled in the art of object oriented programming appreciate that the object only needs to be available to the run time program and can be located anywhere as Applicants teach with respect to all other objects used by their program.

Additionally, in claims 48 and 50, the language "the run time program" was added for consistency with other claims in which it was stated that the run time program, the objects, and the scripts can all be maintained separately.

Finally, in claims 28, 31, 35, and 60 the method language of comprising "the steps of was unintentionally and inadvertently omitted from the claims submitted in the Response of November 28, 2005 although earlier submitted claims had contained the expression. The Response of November 28, 2005 did not note any deletion of the phrase. ...".

IV. Examiner's Comment's

Applicant has proposed a new claim 76. After interacting about the claim the decision not to enter the claim at this time was made. Applicant's can revisit the claim under 3.12 amendment.

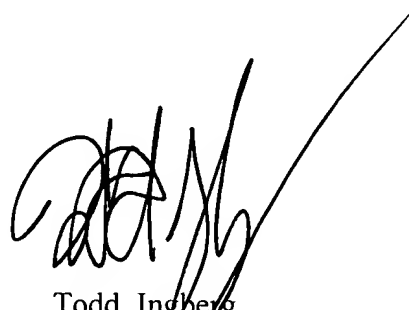
Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Todd Ingberg
Primary Examiner
Art Unit 2193